REMARKS

This amendment is being filed in response to the Office Action having a mailing date of July 6, 2007. Various claims are amended as shown. New claims 31-40 are added. No new matter has been added. With this amendment, claims 1-40 are pending in the application.

I. Initial comments

It is noted that a Preliminary Amendment was filed on May 18, 2004 to submit amendments to the specification, including correction of typographical errors. The filing of said Preliminary Amendment does not appear to have been acknowledged by the present Office Action, although the U.S. Patent Office's Patent Application Information Retrieval (PAIR) system does acknowledge the presence of said Preliminary Amendment in the image file wrapper (IFW).

It is therefore kindly requested that the next communication provide confirmation that said Preliminary Amendment has been entered.

The specification is amended as shown to provide the application serial number of a co-pending patent application.

II. Allowable subject matter

The present Office Action indicated that claims 7-9 and 20 would be allowable if rewritten in independent form. The Examiner is thanked for this indication of allowable subject matter.

New dependent claims 31-34 contain recitations consistent with the subject matter indicated to be allowable. Thus, dependent claims 31-34 would also be allowable.

New independent claim 35 and its dependent claims 36-40 recite at least some of the subject matter indicated to be allowable, and therefore are allowable as well. The appropriate fee authorization to cover payment for the new claims is included with this amendment.

III. Discussion of the other claims and rejections

The present Office Action rejected claims 1-6, 10-19, and 21-30 under 35 U.S.C. 102(e) as being anticipated by Gai (U.S. Patent No. 6,651,096). For the reasons set forth below, these rejections are respectfully traversed.

A. Discussion of independent claim 1

Independent claim 1 recites, *inter alia*, "converting the first rules into minterm representations" and "generating a bit mask for each of the second rules based on their presence in the minterm representations." For example a non-limiting and non-exhaustive embodiment is disclosed in the present application on pages 19-21, wherein multiple bit masks 1001, 1010, 1100, 0111, 1110, 0011, and 1101 are generated for each respective second rule R_{u1}, R_{u2}, etc. based on said rules' presence in the minterm representations M_1, M_2, M_3, and M_4. These minterm representations M_1, M_2, M_3, and M_4 are made up of minterms (products of second/simple rules).

It is respectfully submitted that Gai does not meet the limitations of claim 1. Specifically, Gai discloses a method and apparatus that (1) accesses one or more text-based access control lists (ACLs) and translates them into a Boolean representation in the form of a binary decision diagram (BDD). A Boolean manipulation/transformation engine then (2) performs operations on the ACLs (in BDD format) to generate a single, unified ACL. The Boolean transformation then (3) converts the single, unified ACL from BDD format into a sum of products (SOP) format. Gai describes this method/apparatus on column 3, line 55 to column 4, line 25 reproduced in part below:

> "According to the invention, an ACL converter comprises a boolean transformation engine cooperatively coupled to a boolean manipulation engine for optimizing one or more text-based ACLs for subsequent evaluation by an intermediate network device. The boolean transformation engine accesses the one or more ACLs and translates them into a first boolean representation. In the preferred embodiment, the first

boolean representation is a binary decision diagram (BDD). The boolean manipulation engine then optimizes and merges the ACLs specified for a given interface of the device. That is, the boolean manipulation engine performs one or more operations on the specified ACLs (in BDD format) to generate a single, unified ACL for the given interface .... Preferably, the boolean transformation engine converts the single, unified ACL from BDD format into a second boolean representation, which, in the preferred embodiment, is a sum of products (SOP) format. The single, unified ACL (in SOP format) is then mapped to that portion of the CAM associated with the given interface."

Page 2 (section 2) of the present Office Action has asserted that column 8, lines 7-49 of Gai allegedly meets the limitations of claim 1 that specify "generating a bit mask for each of the second rules based on their presence in the minterm representations." This assertion is respectfully traversed. There is no disclosure, teaching, or suggestion in this cited passage of Gai that involve generating a bit mask for each of the second rules based on their presence in the minterm representations. Column 8, lines 7-49 of Gai is reproduced in full below (emphasis ours):

"For example, the network administrator may assign ACL 416a (ACL 101) to interface 402a for purposes of input security control. Accordingly, upon receipt of a network message at interface 402a, it is compared with ACE statements 502-514 of ACL 416a. The matching is preferably performed logically as a series of sequential steps starting with the first ACE and moving, one ACE at a time, toward the last ACE in the ACL. Once a match is located, the corresponding action is returned and the processing stops. That is, no additional ACEs are examined. If a match is made with an ACE statement having a 'permit' action (e.g., ACE 502), the packet is forwarded. If a match is made with an ACE statement having a 'deny' action (e.g., ACE 506), the packet is dropped. If the matching action is 'permit and log', then the respective message is forwarded and an entry is made in a conventional message log. Similarly, if the matching action is 'deny and log', then the respective message is dropped and a log entry made. If no ACE of the subject ACL matches the message, an implicit action located at the end of the ACL is typically returned (e.g., permit or deny).

The value 'x' of ACLs 416a-416e represents a wildcard. That is, the specified action is independent of the value of 'x' of the message being tested. As shown in ACL 416a, a message having a source address of 1.2.3.3 specified in its IP SA field 104 (FIG. 1) is always forwarded, regardless of the values contained in its IP DA field 108, source port field 202 (FIG. 2), destination port field 204 or the protocol field 104.

ACL 416b (ACL 202), which includes ACE statements 530-536, may be used to implement output security at one or more interfaces. Here, if a packet matches an ACE statement having a permit action, then the packet is forwarded from the interface. If the matching action is deny, then the packet is dropped (i.e., it is not forwarded from the interface). ACLs 416c-416e may be utilized by the network administrator to implement additional features at interface 402a. For example, ACL 416c (ACL 303), which includes ACE statements 538-544, may be associated with an encryption function. If a packet matches an ACE statement of ACL 416c having a 'permit' action, then the packet is encrypted. If the packet matches an ACE having a 'deny' action, then the packet is not encrypted."

From the above-cited passage of Gai, the closest thing that could conceivably be considered as a "mask" are the "x" values in the ACLs shown in his Figures 5A-5B. However, it is abundantly clear that said "x" values are "wildcards" corresponding to data in the ACL itself, and are not generated based on the second rules' "presence in the minterm representation" as recited in claim 1. That is, Figure 5A-5B and the cited passage of Gai pertain to item (1) above that involves accessing (and description of content) of an ACL itself, and does not even involve and has not yet performed item (3) to convert a single, unified ACL from BDD format into a sum of products (SOP) format. In other words, there are not yet any minterm representations in Figure 5A-5B and the cited passage of Gai, since he has not yet performed at that point the item (2) conversion into the Boolean decision diagram (BDD) and the item (3) conversion into SOP format having the products. Thus, it is not possible for Figure 5A-5B and the cited passage of Gai to meet the limitations of claim 1 that require generating a bit mask for each of the second rules "based on their presence in the minterm reprsentations," since minterm representations have not yet been generated by Gai at that point in his process.

Column 15, lines 15-57 of Gai has been additionally cited by the present Office Action as meeting the "generating a bit mask for each of the second rules based on their presence in the minterm representations" limitations of claim 1. This assertion by the present Office Action is respectfully traversed as well. That is, it appears that Gai provides a row in his TCAM 410 for each product term of the SOP-formatted unified ACL. However and in contrast, claim 1 specifies a bit mask for each of the second rules, rather than a bit mask for each minterm representation, which would be the more accurate interpretation of Gai.

In detail with respect to Gai, he describes his TCAM 410 in column 7, lines 5-14 reproduced below (emphasis ours) and elsewhere:

> "It should be understood that a Dynamic CAM, which also has the ability to store 'don't cares' may alternatively be used. Those skilled in the art will also recognize that the same function can be achieved with two Static CAMs, which provide four possible states. To implement the 'don't care' value, the TCAM 410 may be segregated into blocks of cells (each cell being either asserted or de-asserted) such that each block has a corresponding mask that determines whether the particular cells of its block are 'care' or 'don't care'."

Gai then describes how the product terms of the SOP-formatted unified ACL are mapped/stored to each row of the TCAM 410, in column 15, line 63 to column 16, line 17 below (emphasis ours):

> "The ACL converter 424 then maps the SOP-formatted, unified ACL into the TCAM 410, as indicated at block 928. More specifically, the ACL converter maps the SOP-formatted, unified ACL to that portion (e.g., portion 410a) of the TCAM 410 that corresponds to the interface and direction being processed (e.g., interface 402a). Each row of the TCAM 410 preferably corresponds to a product term of the SOP-formatted, unified ACL. For each product term, the ACL converter 424 also stores the corresponding action in memory 411.
>
> More specifically, each row of the TCAM 410 corresponds to an address, and, as described above, memory 411 includes a corresponding space or location for each address (i.e., row) of the TCAM 410. Thus, when the ACL converter 424 writes a particular product term into a row of the TCAM 410, it also writes the action that matches that product term, and is to be implemented by the forwarding entity 404 (e.g., permit, deny, forward to CPU 406, etc.) into the space or location of memory 411 that corresponds to the respective row of the TCAM 410. This process is then repeated for each of the remaining interfaces 402b-402e to which multiple ACLs have been assigned by the network administrator."

Thus from the above, it is abundantly clear that Gai writes his product terms (minterms) into each respective row of his TCAM 410, wherein each row of the TCAM 410 can be provided masking "don't care" values. This implementation of Gai is different from claim 1 that recites "generating a bit mask for each of the second rules based on their presence in the minterm representations."

For example, Gai writes product terms (minterms) into respective rows of his TCAM 410 (which can be masked), and his product terms are inherently made up of simpler individual rules. In other words, Gai can provide masking for his product terms (minterms) themselves, but mentions nothing about generating masks for the individual rules themselves that make up his product terms (minterms). Accordingly, Gai does not meet the limitations of claim 1 that require "generating a bit mask for each of the second rules"

As another example, Gai is silent as to what criteria he uses (if any) to determine whether to mask or not mask groups of rows having product terms (minterms) stored therein, individual ones of said rows, or portions thereof with "don't care" values. In contrast, claim 1 recites generating a bit mask <u>based on</u> the second rules "<u>presence</u> in the minterm representation." Gai does not disclose, teach, or suggest these limitations.

Thus, claim 1 is allowable over Gai.

B. <u>Discussion of the other independent claims</u>

Independent claim 10 as presently amended recites, *inter alia*, "storing a bit mask generated <u>for each simpler rule</u> <u>based on the minterm representations</u> of the complex rules." Independent claim 15 as amended recites, *inter alia*, "at least one <u>minterm</u> … made of a plurality of <u>second rules</u>" and "storing a bit mask <u>for each of the second rules</u>." Independent claim 19 recites, *inter alia*, "convert the first rules into <u>minterm</u> representations" and "generate a bit mask <u>for each of the second rules</u> <u>based on their presence in the minterm representations</u>." Independent claim 22 recites, *inter alia*, "a means for converting the complex rules into <u>minterm</u> representations" and "a means for generating a bit mask <u>for each simpler rule</u> <u>based on the minterm representations</u> of the complex rules." Independent claim 26 recites, *inter alia*, "a first rule reduced into at least one <u>minterm</u> <u>made of a plurality of second rules</u>" and "a bit mask generated <u>for each of the second rules</u>."

Because Gai does not disclose, teach, or suggest these limitations, said claims are allowable.

C. Other claim discussions

New dependent claims 31-34 recite features related to the bit masks. As previously explained above, Gai appears to be silent as to the criteria or methodology (if any) he uses to provide masks for his TCAM 410. In contrast, claims 31-34 recite enabled/disabled bit positions in the bit mask based on the presence or non-presence of a rule in the minterm (or minterm representations). Thus, claims 31-34 are alloawble.

Various amendments are made to the claims as shown to more precisely recite the subject matter contained therein, to provide consistent antecedent basis, to remove unnecessary language, and/or to otherwise place such claims in better form and in a manner consistent with the specification.

IV. Information disclosure statement (IDS)

An IDS having references listed therein, the appropriate fee authorization, and copies of non-published U.S. patent reference(s) listed in the IDS are being submitted along with this amendment. It is kindly requested that an Examiner-initialed copy of the IDS be returned along with the next communication, so as to confirm that the references listed therein have been entered and considered.

V. Conclusion

If the applicant's attorney (Dennis M. de Guzman) has overlooked a teaching in any of the cited references that is relevant to the allowability of the claims, the Examiner is requested to specifically point out where such teaching may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact Mr. de Guzman at (206) 622-4900.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC

/Dennis M. de Guzman/

Dennis M. de Guzman
Registration No. 41,702

DMD:sc

701 Fifth Avenue, Suite 5400
Seattle, Washington 98104
Phone: (206) 622-4900
Fax: (206) 682-6031

998748_1.DOC